This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Currently Amended) A method comprising:

~~for~~ configuring a computer to generate computer executable instructions, using object-based computer code of an object-oriented programming language~~, the method comprising:~~

utilizing an explicit interface member mechanism that enables a class to implement an explicit interface member by explicitly specifying the relationship between the class and the explicit interface member, wherein the explicit interface member mechanism enables an implemented explicit interface member to be excluded from a public interface of said class; and

storing said class in a form that includes said implemented explicit interface member in a computer readable storage medium.

2. (Previously presented) A method according to claim 1, wherein said specifying of the relationship includes specifying a qualified name of the class.

3. (Previously presented) A method according to claim 2, wherein said specifying of the qualified name includes specifying an interface name and said at least one interface member name.

4-5. (Canceled)

6. (Previously presented) A method according to claim 1, wherein the explicit interface member mechanism enables the class to implement an internal interface not accessible to a consumer of said class.

7. (Original) A method according to claim 1, wherein said explicit interface member mechanism enables disambiguation of a plurality of interface members having the same signature.

8. (Original) A method according to claim 1, wherein said explicit member mechanism enables disambiguation of a plurality of interface members having the same signature and return type.

9. (Original) A method according to claim 1, wherein in addition to allowing the implementation of public interface members, said explicit interface member mechanism enables the implementation of private interface members.

10. (Original) A method according to claim 1, wherein said explicit interface member mechanism enables the implementation of a plurality of non-conflicting specific versions of a generic interface.

11. (Currently Amended) A method according to claim 1, wherein the computer code is programmed according to an object-oriented programming language~~, and said object-oriented programming language is one of C#, Fortran, Pascal, Visual Basic, C, C++ and Java~~.

12. (Original) A method according to claim 1, wherein an implementation of an explicit interface member is a method, property, event, or indexer declaration that references a fully qualified interface member name.

13. (Previously presented) A method according to claim 1, wherein the class names an interface in a base class list of the class that contains a member whose fully qualified name, type, and parameter types exactly match those of the implementation of the explicit interface member.

14. (Previously presented) A method according to claim 1, wherein said explicit interface member mechanism includes an interface mapping mechanism that locates implementations of interface members in the class.

15. (Previously presented) A method according to claim 14, wherein said interface mapping mechanism locates an implementation for each member of each interface specified in a base class list of the class.

16. (Canceled)

17. (Original) A method according to claim 1, wherein it is not possible to override an explicit interface member implementation, but where an explicit interface member implementation calls another virtual method, derived classes are capable of overriding the implementation.

18. (Previously presented) A method according to claim 1, wherein the class inherits an interface implementation is permitted to re-implement the interface by including the interface in the base class list of the software component.

19. (Original) A method according to claim 1, wherein said explicit interface member mechanism prevents conflict among specific implementations of a generic interface.

20-22. (Canceled)

23. (Currently Amended) A computer readable storage medium including processor-executable instructions, the processor-executable instructions generated from~~, in a computer system, having stored thereon~~ a plurality of ~~computer-executable~~ modules written in an object-oriented programming language, the computer executable modules comprising:

an explicit interface member mechanism that enables a class to implement an explicit interface member by explicitly specifying the relationship between the class and an interface member, wherein said explicit interface member mechanism enables an implemented explicit interface member to be excluded from a public interface of said class.

24. (Previously presented) A computer readable storage medium according to claim 23, wherein said specifying of the relationship includes specifying a qualified name of the at least one software component.

25. (Previously presented) A computer readable storage medium according to claim 24, wherein said specifying of the qualified name includes specifying at least one interface name and said at least one interface member name.

26. (Previously presented) A computer readable storage medium according to claim 23, wherein said explicit interface member mechanism enables an explicit interface member implementation to be excluded from the public interface of said class.

27. (Canceled)

28. (Previously presented) A computer readable storage medium according to claim 23, wherein the explicit interface member mechanism enables the class to implement an internal interface not accessible to a consumer of said class.

29. (Previously presented) A computer readable storage medium according to claim 23, wherein said explicit interface member mechanism enables disambiguation of a plurality of interface members having the same signature.

30. (Previously presented) A computer readable storage medium according to claim 23, wherein said explicit member mechanism enables disambiguation of a plurality of interface members having the same signature and return type.

31. (Previously presented) A computer readable storage medium according to claim 23, wherein in addition to allowing the implementation of public interface members, said explicit interface member mechanism enables the implementation of private interface members.

32. (Previously presented) A computer readable storage medium according to claim 23, wherein said explicit interface member mechanism enables the implementation of a plurality of non-conflicting specific versions of a generic interface.

33. (Currently Amended) A computer readable storage medium according to claim 23, wherein the object-oriented programming language is ~~one of~~ C#~~, Fortran, Pascal, Visual Basic, C~~ and ~~C++ and Java~~.

34. (Previously presented) A computer readable storage medium according to claim 23, wherein an implementation of an explicit interface member is a method, property, event, or indexer declaration that references a fully qualified interface member name.

35. (Previously presented) A computer readable storage medium according to claim 23, wherein said the class names an interface in the base class list of the class that contains a member whose fully qualified name, type, and parameter types exactly match those of the implementation of the explicit interface member.

36. (Previously presented) A computer readable storage medium according to claim 23, wherein said explicit interface member mechanism includes an interface mapping mechanism that locates implementations of interface members in said class.

37. (Previously presented) A computer readable storage medium according to claim 36, wherein said interface mapping mechanism locates an implementation for each member of each interface specified in a base class list of the class.

38. (Canceled)

39. (Previously presented) A computer readable storage medium according to claim 23, wherein it is not possible to override an explicit interface member implementation, but where an explicit interface member implementation calls another virtual method, derived classes are enabled to override the implementation.

40. (Previously presented) A computer readable storage medium according to claim 23, wherein the class inherits an interface implementation is permitted to re-implement the interface by including it in a base class list of the class.

41. (Previously presented) A computer readable storage medium according to claim 23, wherein said explicit interface member mechanism prevents conflict among specific implementations of a generic interface.

42.-60. (Canceled)

61. (Previously presented) A method of generating an object comprising:

receiving, by a compiler, source code identifying a class that implements an interface and a member, the class specifying a relationship between the member and the name of the interface;

implementing, by the compiler, the member as an explicit interface member in the class in response to detecting the relationship between the member and the name of the interface, wherein the explicit interface member is excluded from a public interface of the class and the explicit interface member is accessible from the interface; and

initializing an instance of said class in a computer readable storage medium.

62. (Previously presented) The method of claim 61, wherein said specifying a relationship includes specifying a qualified name of the class.

63. (Previously presented) The method of claim 62, wherein said specifying a qualified name includes specifying an interface name and said interface member name.

64. -65.(Canceled)

66. (Previously presented) The method of claim 61, wherein said class implements an internal interface that is not accessible to a consumer of said software component.

67. (Previously presented) The method of claim 61 wherein said implemented explicit interface member enables disambiguation of interface members with the same signature.

68. (Previously presented) The method of claim 61 wherein said implemented explicit interface member enables disambiguation of interface members with the same signature and return type.

69. (Previously presented) The method of claim 61 wherein said implemented explicit interface member enables the implementation of private interface members.

70. (Previously presented) The method of claim 61, wherein said implemented explicit interface member enables a plurality of non-conflicting specific versions of a generic interface.

71. (Previously presented) The method of claim 61, wherein the explicit interface member is implemented in an object-oriented programming language.

72. (Previously presented) The method of claim 61, wherein said implemented explicit interface member is a member selected from a group of members consisting of a method, a property, an event, and an indexer declaration that references a fully qualified interface member name.

73. (Previously presented) The method of claim 61, wherein said class names an interface in a base class list of the class that contains a member whose fully qualified name, type, and parameter types exactly match those of the implemented explicit interface member.

74. (Previously presented) The method of claim 61 further comprising:
locating implemented interface members in each interface in said class.

75. (Previously presented) The method of claim 66, wherein said locating implementations of each interface member includes locating an implementation for each member of each interface specified in a base class list of the software component.

76. (Canceled)

77. (New) The method of claim 61, wherein it is not possible to override an explicit interface member implementation, but where an explicit interface member implementation calls another virtual method, derived classes are enabled to override the implementation.

78. (New) The method of claim 61, wherein said class that inherits an interface implementation is permitted to re-implement the interface by including it in the base class list of the class.

79. (New) The method of claim 61, wherein said implementing said explicit interface member prevents conflict among specific implementations of a generic interface.